PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :
H04Q 11/00, 11/04, H04J 3/14, 3/24

A1

(11) International Publication Number: WO 00/13455

(43) International Publication Date: 9 March 2000 (09.03.00)

(21) International Application Number: PCT/US99/17229

(22) International Filing Date: 30 July 1999 (30.07.99)

(30) Priority Data:
09/141,136 27 August 1998 (27.08.98) US

(71) Applicant: INTEL CORPORATION [US/US]; 2200 Mission College Boulevard, P.O. Box 58119, Santa Clara, CA 95052–8119 (US).

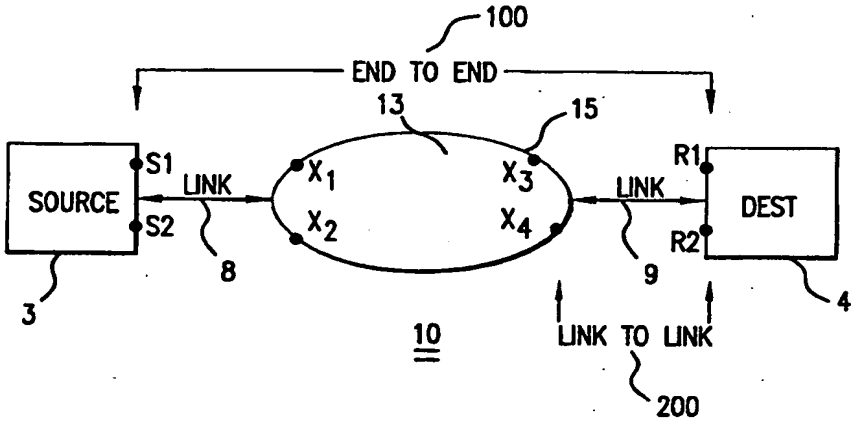
(72) Inventors: DROTTAR, Ken; 10423 NW Royal Rose Court, Portland, OR 94229 (US). DUNNING, David, S.; 14055 NW Evergreen Street, Portland, OR 97229 (US). CAMERON, Donald, F.; 3435 N.W. Vaughan, Portland, OR 97210 (US).

(74) Agents: HELLER, Paul, H. et al.; Kenyon & Kenyon, Suite 700, 1500 K Street, N.W., Washington, DC 20005 (US).

(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD AND APPARATUS FOR INPUT/OUTPUT LINK RETRY, FAILURE AND RECOVERY IN A COMPUTER NETWORK



(57) Abstract

Method for transmitting data from a source (3) to a destination (4) node including an intermediary point and packet sequence number. A copy of the packet is stored in a buffer at the source node until receiving an acknowledgment that it was successfully received by the intermediary point. The intermediate point assigns an intermediate point sequence number to the packet and a copy of the packet is retained in a buffer until receiving an acknowledgment from the next delivery point. The packet in the buffer is de–allocated once it is successfully received. Upon receipt of an error indication, each packet is retransmitted. At the receiving end, all received packets following the packet associated with the error indication are dropped until successfully receiving a retransmitted version of the packet. A single negative acknowledgment indicates that the packet associated with the negative acknowledgment includes at least one error and to simultaneously indicate that all previous packets received prior to the packet associated with the negative acknowledgment were received correctly. An independent link sequence number is assigned to each packet before transmitting.

METHOD AND APPARATUS FOR INPUT/OUTPUT LINK RETRY, FAILURE AND RECOVERY IN A COMPUTER NETWORK

RELATED APPLICATIONS

This application claims priority to U.S. Provisional Application No. 60/057,221 filed on August 29, 1997, entitled "Method and Apparatus for Communicating Between Interconnected Computers, Storage Systems, and Other Input/Output Subsystems" by inventors Ahmet Houssein, Paul A. Grun, Kenneth R. Drottar, and David S. Dunning, and to U.S. Provisional Application No. 60/081,220, filed on April 9, 1998, entitled "Next Generation Input/Output" by inventors Christopher Dodd, Ahmet Houssein, Paul A. Grun, Kenneth R. Drottar, and David S. Dunning. These applications are hereby incorporated by reference as if repeated herein in their entirety, including the drawings. Furthermore, this application is related to U.S. Patent Application No. 09/141,151 filed by David S. Dunning and Kenneth R. Drottar on even date herewith and entitled "Method and Apparatus for Controlling the Flow of Data Between Servers." This application is also related to U.S. Patent Application No. 09/141,134 filed by David S. Dunning, Ken Drottar and Richard Jensen on even date herewith and entitled "Method and Apparatus for Controlling the Flow of Data Between Servers Using Optimistic Transmitter."

BACKGROUND OF THE INVENTION

The present invention relates generally to methods and apparatuses for controlling the flow of data between nodes (or two points) in a computer network, and more particularly to a method and apparatus for controlling the flow of data between two nodes (or two points) in a system area network.

For the purposes of this application, the term "node" will be used to describe either an origination point of a message or the termination point of a message. The term "point" will be used to refer to a transient location in a transmission between two nodes. The present invention includes communications between either a first

In an architecture that permits large data packets, unnecessarily retransmitting excess packets can become a significant efficiency concern. For example, retransmitting an entire window of data packets, each on the order of 4 Gigabytes, would be relatively inefficient.

Other known flow control protocols require retransmission of only the packet received in error. This requires the receiver to maintain a buffer of the correctly received packets and to reorder them upon successful receipt of the retransmitted packet. While keeping the bandwidth requirements to a minimum, this protocol significantly complicates the receiver design as compared to that required by Go Back *n* ARQ.

The present invention is therefore directed to the problem of developing a method and apparatus for controlling the flow of data between nodes in a system area network that improves the efficiency of the communication without overly complicating the processing at the receiving end.

SUMMARY OF THE INVENTION

The present invention provides a method for transmitting data in a network from a source node to a destination node. According to the method of the present invention, data packets are transmitted from the source node to at least one intermediary point. Each of the packets is assigned a corresponding sequence number by the source node. A copy of each packet is retained in a buffer at the source node until an acknowledgment is received that the packet was successfully received by an intermediary point. At the intermediary point, an intermediate point sequence number is assigned to each packet received by the intermediary point.

The present invention provides an apparatus for communicating data between two links of a fabric made of multiple links. The apparatus includes two switches and a buffer. The first switch is disposed in a first point of a link and transmits the data packets from the first point in the link to a second point in the link. The first switch assigns first point sequence numbers to the packets, which first point sequence numbers are independent from source sequence numbers assigned by a source of the packets. The buffer is disposed in the first point, is coupled to the first

of connecting I/O devices to a computer node, or connecting two computer nodes together, based on load/store memory transactions. An interconnect based on I/O pass through is transparent to the entities at either end of the interconnect. NG I/O (physical) is a minimum set of wires and the protocol that runs on the link that interconnects two entities. For example, the wires and protocol connecting a computer node to a switch comprise a link. NG I/O bundled refers to the capability to connect two or more NG I/O links together in parallel. Such bundled links can be used to gain increased bandwidth or improve the overall reliability of a given link. According to the present invention, a switch is defined as any device that is capable of receiving packets (also referred to as I/O packets) through one or more ports and re-transmitting those packets through another port based on a destination address contained in the packet. In network terms, a switch typically operates at the data link layer of the Open Systems Interconnection (OSI).

FIG 1 illustrates the overall NG I/O link architecture according to an exemplary embodiment of the present invention. The overall NG I/O link architecture can be illustrated as including one or more computers 210 (e.g., servers, workstations, personal computers, or the like), including computers 210A and 210B. The computers 210 communicate with each other via a switched NG I/O fabric that may include a layered architecture, including a network layer 212, a data link layer 214 and a physical layer 216. An NG I/O switch 220 (e.g., including data link and physical layers) interconnects the computers 210A and 210B. Each computer 210 can communicate with one or more I/O devices 224 (224A and 224B) via the NG I/O fabric using, for example, an I/O pass through technique 226 according to the present invention and described in greater detail below. Each computer 210 can communicate with one or more I/O devices 224 (224A and 224B), alternatively using a distributed message passing technique (DMP) 227. As a result, I/O devices 224 may be remotely located from each computer 210.

FIG 2 is a block diagram of an NG I/O architecture for I/O pass through according to an embodiment of the present invention. The NG I/O architecture includes a computer 310 and a computer 360, each which may be a server, workstation, personal computer (PC) or other computer. Computers 310 and 360

While the embodiment of the NG I/O architecture of the present invention illustrated in FIG 2 includes an NG I/O to PCI bridge 320, it should be understood by those skilled in the art that other types of bridges can be used. For example, generically speaking, bridge 320 can be referred to as a "network to peripheral bridge" for converting network packets to and from a format that is compatible with bus 322 (bus 322 may be a wide variety of types of I/O or peripheral buses, such as a PCI bus). Likewise, PCI storage controller 324 can be generically referred to as a "peripheral storage controller" for any of several types of I/O devices. Therefore, the present invention is not limited to PCI bridges, but rather, is applicable to a wide variety of other I/O buses, such as Industry Standard Architecture (ISA), Extended Industry Standard Architecture (EISA), Accelerated Graphics Port (AGP), etc. PCI is merely used as an example to describe the principles of the present invention. Similarly, NG I/O to host bridge 364 can be generically referred to as a "network to host bridge" because it converts (NG I/O) network packets to and from a host format (host transactions).

FIG 2 illustrates that an NG I/O fabric 328 can be used to move storage devices out of the server cabinet and place the storage devices remote from the computer 310. Fabric 328 can include one or more point-to-point links between computer 410 and each I/O system 418, or can include a number of point-to-point links interconnected by one or more switches. This architecture permits a more distributed environment than presently available.

The present invention provides a simple means to create a working network with flow control mechanisms which do not allow for lost data due to congestion and transient bit errors due to internal or external system noise. The present invention uses an approach to flow control that does not require end-to-end or link-to-link credits, rather the present invention combines the ability to detect a corrupted or out of order packets and retry (resend) any/all packets to maintain that all data is delivered uncorrupted, without losing any data and in the order that the data was sent. This is accomplished by assigning a sequence number and calculating a 32 bit Cyclic Redundancy Check (CRC) with each packet and acknowledging (ACK) or negative acknowledging (NAK) each packet.

sequence number associated by link. On any given link, packets must arrive in the order transmitted. On any given link, descriptors are retried in the order they were queued. Note, that as an efficiency improvement to this algorithm, the link-switch for link 8 can ACK multiple packets at one time by ACKing the highest sequence number that has been correctly received, e.g., if the source 3 receives an ACK for packet #9, then receives an ACK for packet #14, packets #10-#13 are also implicitly ACKed. After the link-switch for link 8 sends an ACK that the packets have been sent correctly, link-switch for switch 9 sends the packets to destination 4. Destination 4 must ACK send back an acknowledgment to link-switch for switch 9 that the data was sent correctly. A new set of sequence numbers is assigned to the packets sent from link-switch for switch 9 to the destination.

Transient errors are errors that occur when packets are sent from a sending node to a receiving node. In the event of a transient error due to internal or external system noise, data may be corrupted between the source 3 and the destination 4. The receiver of the packets must calculate the CRC across the data received, and compare it to the CRC appended to the end of the packet. If the calculated CRC and the received CRC match, the packet will be ACKed. If the two CRC's do not match, that packet must be NAKed, again identified by the sequence number. Upon receipt of a NAK, the sender must resend the specified packet again, followed by all packets following that packet. For example, if the sender has sent packets up to sequence number 16 but receives a NAK for packet #14, it must resend packet #14, followed by packet #15 and packet #16. Note that ACKs and NAKs can still be combined. Using the example in the previous paragraph, of packet 9 is ACKed, then packets #10-#13 are assumed received in order and without data corruption, followed by packet #14 with corrupted data; a NAK of packet#14 signifies that packets #10 - #13 were received without error, but that packet #14 was received with error and must be resent.

FIG 4 is a block diagram illustrating NG I/O links according to an embodiment of the present invention. Fabric 400 is connected between nodes A, B and C labeled 401, 402 and 403, respectively. As shown in FIG 4, a link 411 is disposed between node A and fabric 400, a link 412 is disposed between node B and

alternative, if a NAK has been received by the switch from node B, then the switch determines which packets must be resent. According to the features of the present invention, new sequence numbers 101-106 are used to identify the packets. Thus, a NAK for sequence number 104 signifies that packets represented by sequence numbers 101-103 were received without error but the packets represented by sequence numbers 104-106 was received with error and must be resent.

If congestion in the network occurs, received packets may not be able to immediately make progress through the network. Congestion in a network is the overcrowding of packets across the network. Congestion control and congestion management are two mechanisms available for a network to effectively deal with heavy traffic volumes. Referring back to FIG 3, when a local buffer space is filled at a receiving queue, additional packets will be lost, e.g., when queue X1 fills up, packets that follow will be thrown away. However, given that retry can occur across each point of a network instead of each end, packets being thrown away are relatively simple to recover from. As soon as receiving queue X1 starts moving packets out of its receive buffers, it opens up room for additional packets to be received. The receive queue X2 will check the sequence number of the next packet it receives. In the event that source 3 has sent packets that were dropped, the first dropped packet will be NAKed and therefore resent from that packet on.

According to the present invention, the send queue S1 just keeps sending packets until its send queue is full of packets that have not been ACKed. It must wait for an ACK for those packets before it can reuse those buffers (it needs to be able to retry those packets if necessary).

There are many advantages of the present invention. For example, the present invention allows for retry of corrupted packets at each point in the network instead of at the source and the destination of the network. According to the present invention, after a first node (source) transmits to and receives an acknowledgment from an intermediate point the first node is no longer relied upon to resend information that may be corrupted later in the transmission path from other intermediary nodes to the destination. Thus, the retry feature of the present invention simplifies data transmission and makes it more efficient. Since the first

WHAT IS CLAIMED IS:

1. A method for transmitting data in a network from a source node to a destination node comprising the steps of:

a) transmitting data in a plurality of packets from said source node to at least one intermediary point, said plurality of packets being assigned a corresponding sequence number;

b) retaining a copy of each packet in a buffer at said source node until receiving an acknowledgment that said each packet was successfully received by said intermediary point; and

c) assigning an intermediate point sequence number to each packet received by the intermediate point.

2. The method according to claim 1, further comprising the step of retaining a copy of each packet in a buffer at said intermediate point until receiving an acknowledgment that said each packet was successfully received.

3. The method according to claim 1, further comprising the steps of:

d) de-allocating a particular packet in the buffer at the source node upon receipt of an acknowledgment associated with said particular packet from said intermediary node; and

e) de-allocating any other packets in the buffer between said particular packet and a last acknowledged packet.

4. The method according to claim 1, further comprising the steps of:

d) retransmitting said each packet and all subsequent packets upon receipt of an error indication; and

e) dropping all received packets following said each packet associated with the error indication until successfully receiving a retransmitted version of said each packet.

9. An apparatus for communicating data between two links of a fabric including a plurality of links, said apparatus comprising:

a) a first switch being disposed in a first point of a link and transmitting the data in a plurality of packets from the first point in the link to a second point in the link; said first switch assigning first point sequence numbers to the plurality of packets, said first point sequence numbers being independent from source sequence numbers assigned by a source of the packets;

b) a buffer being disposed in the first point, being coupled to the first switch and storing each packet until receiving either an acknowledgment that said each packet was successfully received or an error indication that a received version of said each packet included at least one error; and

c) a second switch being disposed in the second point, receiving each of the transmitted data packets, and upon receipt of an error free packet sending an acknowledgment to indicate successful receipt of said error free packet and all packets in sequence between a last acknowledged packet and said error free packet, said second switch assigning second point sequence number to said received packets, said second point sequence numbers being independent from said first point sequence numbers.

10. The apparatus according to claim 9, wherein the first switch de-allocates a packet in the buffer upon receipt of an acknowledgment associated with said packet in the buffer in addition to all packets preceding said packet in the buffer.

11. The apparatus according to claim 9, wherein the first switch retransmits a particular packet and all packets in sequence subsequent to the particular packet upon receipt of an error indication associated with said particular packet.

12. The apparatus according to claim 11, wherein said second switch drops all received packets in sequence following said particular packet until successfully receiving a retransmitted version of said particular packet.

e) dropping all received packets following said each packet associated with the error indication until successfully receiving a retransmitted version of said each packet.

17. A program storage device readable by a machine, tangibly embodying a program of instructions executable by a machine to perform method steps for transmitting data between switches in a fabric having a plurality of links, said method comprising the steps of:

a) transmitting the data in a plurality of packets from a first switch to a second switch, said plurality of packets being assigned a sequence number by the first switch;

b) retaining each packet in a buffer at the first switch until receiving either an acknowledgment indicating that said each packet was successfully received or an error indication that a received version of said each packet included at least one error, while simultaneously transmitting additional packets;

c) using a single negative acknowledgment to indicate that a packet associated with the negative acknowledgment includes at least one error and to simultaneously indicate that all previous packets received prior to the packet associated with the negative acknowledgment were received correctly; and

d) assigning a link sequence number to each of said packets before transmitting each of the packets from a second switch, said link sequence number assigned by the second switch being independent of the link sequence number assigned by the first switch.

18. The device according to claim 17, wherein the method further comprises the step of indicating successful receipt of all packets between a last acknowledged packet and a particular packet by sending a single acknowledgment.

19. The device according to claim 17, wherein the method further comprises the steps of:

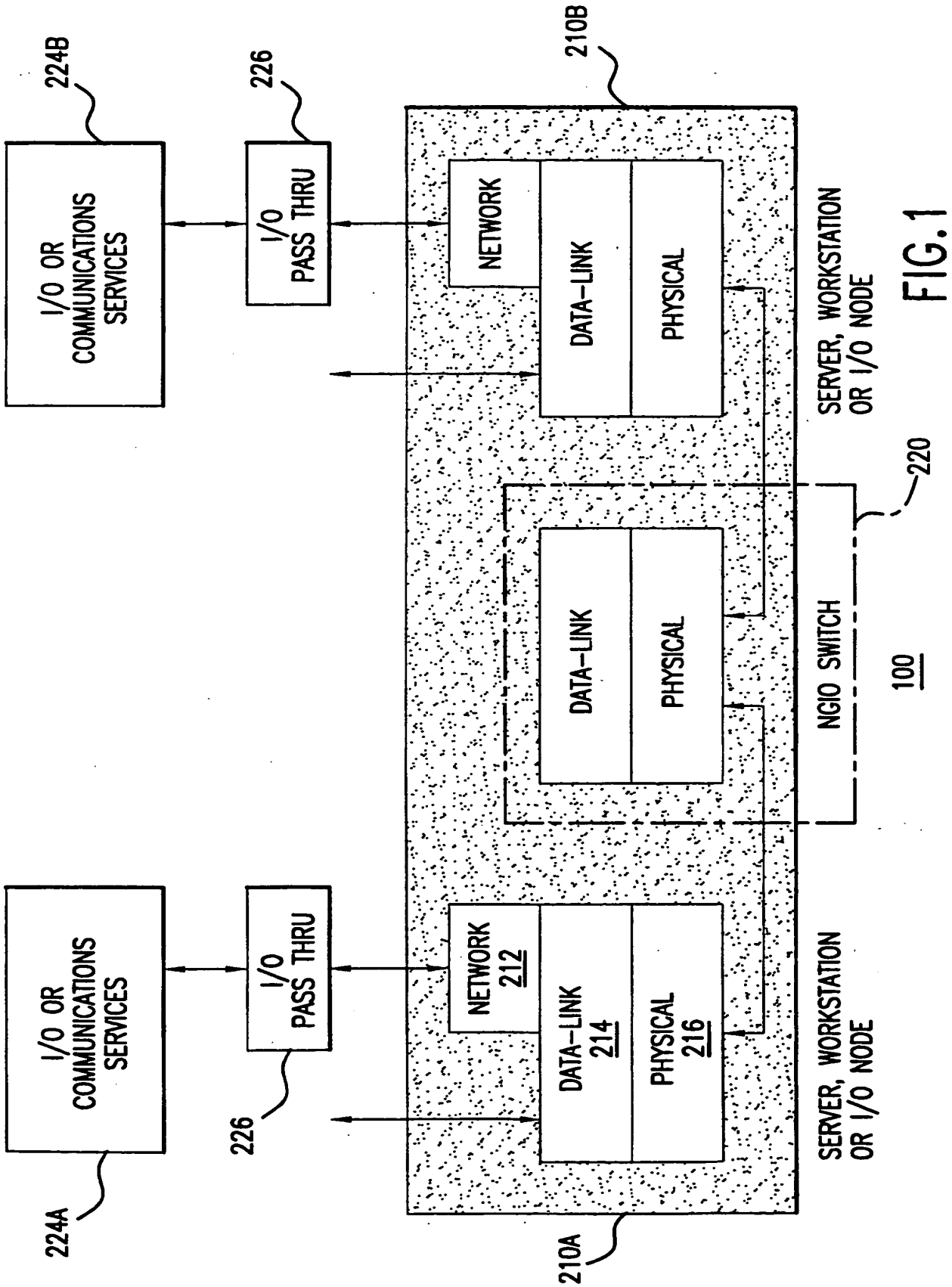

224B
226
210B
I/O OR COMMUNICATIONS SERVICES
I/O PASS THRU
NETWORK
DATA-LINK
PHYSICAL
SERVER, WORKSTATION OR I/O NODE
220
NGIO SWITCH
100
DATA-LINK
PHYSICAL
I/O OR COMMUNICATIONS SERVICES
I/O PASS THRU
NETWORK 212
DATA-LINK 214
PHYSICAL 216
SERVER, WORKSTATION OR I/O NODE
226
224A
210A

FIG. 1

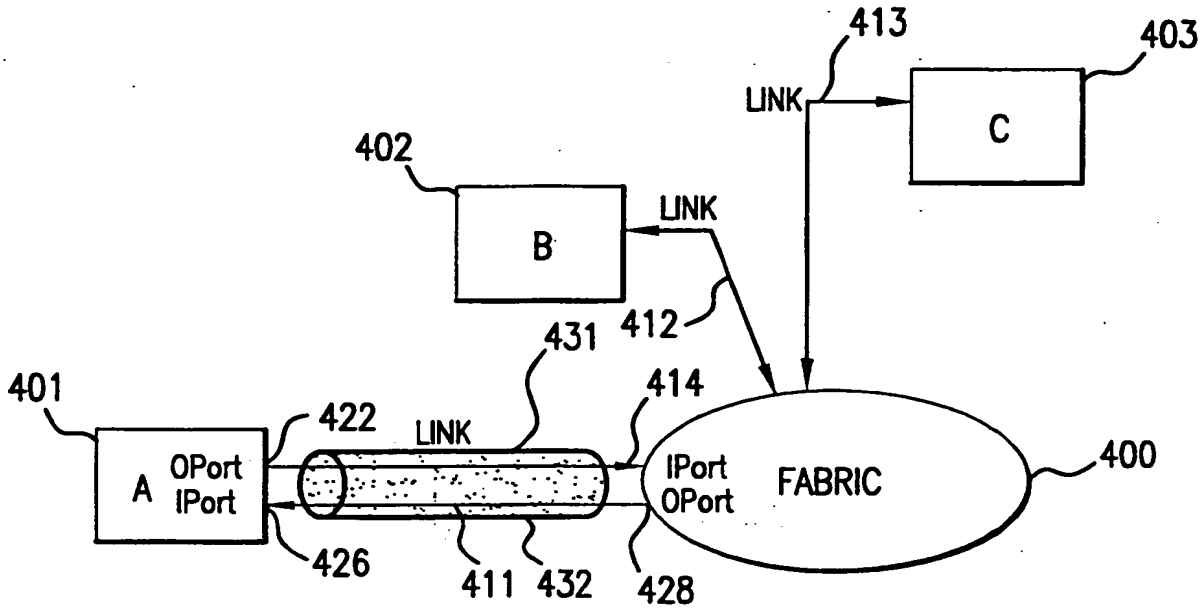

413
403
LINK
C
402
B
LINK
412
431
414
401
422
LINK
A
OPort
IPort
IPort
OPort
FABRIC
400
426
411
432
428

FIG.4

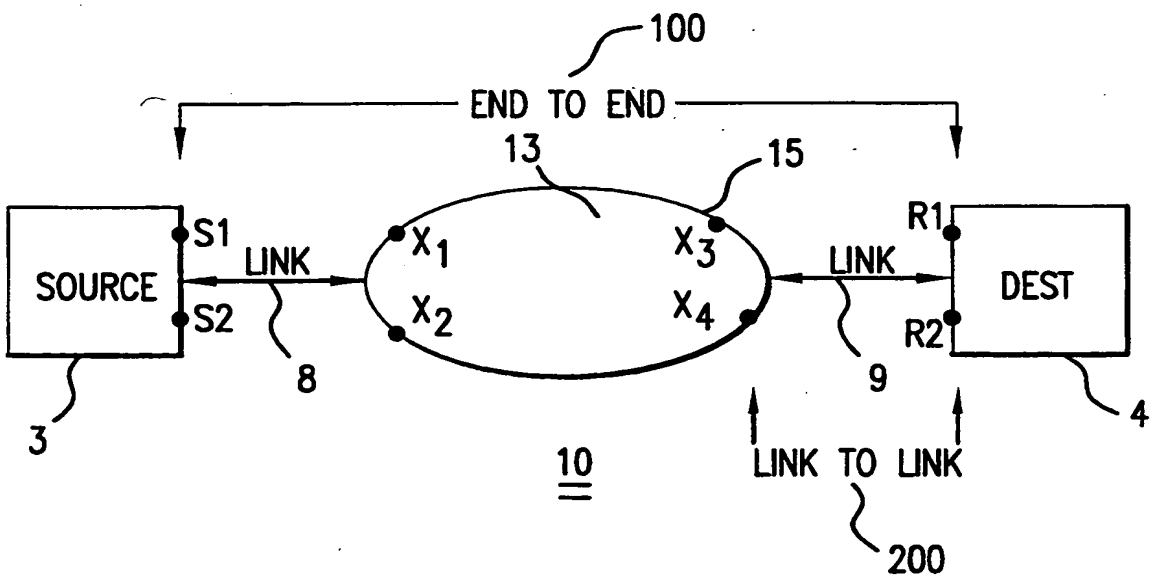

100
END TO END
13
15
S1
R1
SOURCE
LINK
X1
X3
LINK
DEST
S2
X2
X4
R2
8
9
3
4
10
LINK TO LINK
200

FIG.3